Design and Implementation of Embedded Home Gateway Based on Linux

Guang Dong[1], Ming Fang[2], Wei He[3], Yuhang Wang[4]

School of Computer Science and Technology, Changchun University of Science and Technology

Department of Control Science and Engineering, Zhejiang University

[1]light@cust.edu.cn; [2]fangming@cust.edu.cn; [3]hw@cust.edu.cn; [4]Wyh.hello@163.com

Abstract-**This paper introduces the software architecture of home gateway system based on embedded Linux technology, using Socket programming technology and implementing exchange between the RS232 data within the system and external Internet data. This system completes the database design on embedded home gateway using SQLite architecture, and implements the application of the SQLite embedded database on home gateway. Analysis of TCP/ IP protocol, according to the embedded home gateway protocol, by cutting the TCP/ IP protocol layer, the system implements the transplant of TCP/ IP protocol stack on embedded home gateway. Web server is successfully ported, and embedded Linux operation system is ported to home gateway platform.**

Keywords-Embedded Linux; Home Gateway; SQLite; Protocol Stack; Web Server

I. INTRODUCTION

The robustness and stability required by embedded systems are higher than general purpose computing system. Embedded Linux adheres to the robustness, reliability and stability characteristics of Linux. Linux is applied as an embedded operating system to improve the robustness and stability of the embedded system.

The network stack design of Linux kernel is simple and from the practical point of view, and it achieves a more complete set of network protocols. In the high-level network protocols, Linux supports ftp, telnet and rlogin protocols. Linux also provides file access to other machines in the network (NFS, Network File System). Linux provides the most complete support on the most commonly used TCP / IP protocol.

Linux uses C compiler of GNU project, and debugs by using source-level debugger gdb. Through the communication with the serial port and the gdb, C source level debugging can be made, and even other programs can be downloaded to RAM or Flash memory through the serial port. gdb can make all the software / hardware initialization code run until all the core start. Once the core operates normally, the developers can use other more advanced debugging tools such as Kgdb. If it is connected to a network, the more convenient xgdb can be used to debug application program.

Linux itself and many softwares running on it are issued on "CopyLeft" way, and the openness of source codes adapts to the diversity of smart devices in the post-PC era. This makes the embedded products developed by embedded Linux-based low total cost.

Home network central controller establishes hardware platform focus on ARM microprocessor, connected by external broadband Ethernet and internal wireless LAN integration on appliances. The status of appliances is controlled and queried by remote Web browser, local touch screen and telephone voice. Various data storage and management need a back-end database to support [1-3].

Database for embedded system is selected among numerous databases[4]. The characteristics of embedded system development environment determine that the database needs to have the following characteristics: (1) An appropriate volume is needed first when selecting database for embedded application because data storage and program execution in embedded system generally has greater space limits. (2)There are many applications in embedded development, and various user requirements need a fully functional database to realize data management. For the developers, complete development documents of adopted database technology and developmental ease are also required. (3)As a product development, open source code can reduce production costs. More importantly, it can provide the most exhaustive resolution for improvement and maintenance of the products.

The design thinking of SQLite is a small, rapid and minimal management [5]. It provides multiple database interfaces. Though its functions are less than Berkeley DB, we do not need complex relevance among tables. SQLite has an ideal balance between sizes and functions, so that complete open source code can be called ideal "embedded database".

The remote monitoring of B/S mode can be realized by writing CGI programs with successfully ported Web servers and databases. The planned database is dynamic web pages CGI program and provides the necessary database support for local data processing program. The whole system realizes remote monitoring for various types of equipments by B/S mode [6-8].

II. SOFTWARE STRUCTURE DESIGN ON EMBEDDED HOME GATEWAY

Embedded Home Gateway mainly realizes the data exchange of major internal RS232 and external internet data. When the gateway receives an access request from the remote host, it reads from the RS-232 serial, does the corresponding processing, calls kernel, BSD socket, the transport layer and network layer by Linux system, and adds the corresponding logical addresses and other data for the network layer. Then IP datagram is encapsulated, a physical address is added to the MAC layer and other corresponding MAC frame data are added to network card. Finally, the data are sent by hardware. The software structure is shown in Fig 1.

A. *Socket Programming in Linux*

Network communication is a basic function of gateway service. The network communication of Linux system is evolved from UlVIX4.3BSD model, which supports all the BSD Socket and TCP / IP functions. Linux supports a wide range of Socket so that it is suitable for both the general inter-process communication and the inter-process communication in network environment. Socket has logically three elements: domain, type and procedures. Domain shows in which kind of network the socket is used; type means the network communication pattern, which includes connection oriented and connectionless two means of communication. Combining with domain and type can identify the applicable procedures in general. The process with socket communication uses server/client mode. Linux kernel provides a unified system call interface sys_socket_call (int call, unsigned long * args) for all the operations related with Socket. The first parameter call is the specific opcode defined in include / linux / net. H. The basic socket functions are shown in Fig 2.

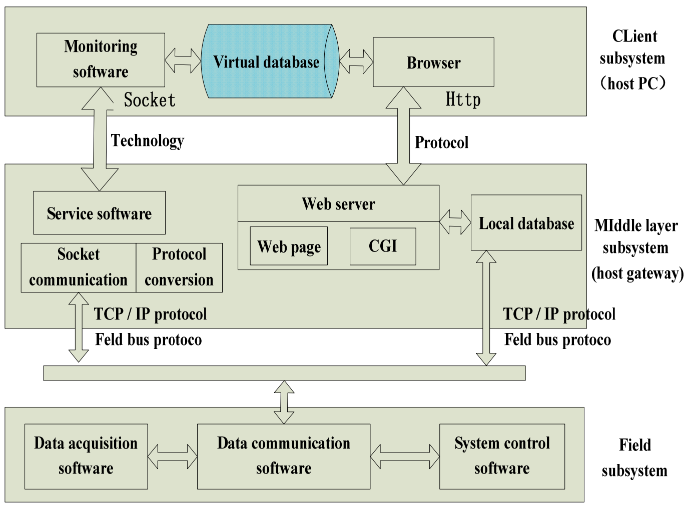

Fig. 1. Software Structure of Embedded Home Gateway

B. *RS -232 Operations*

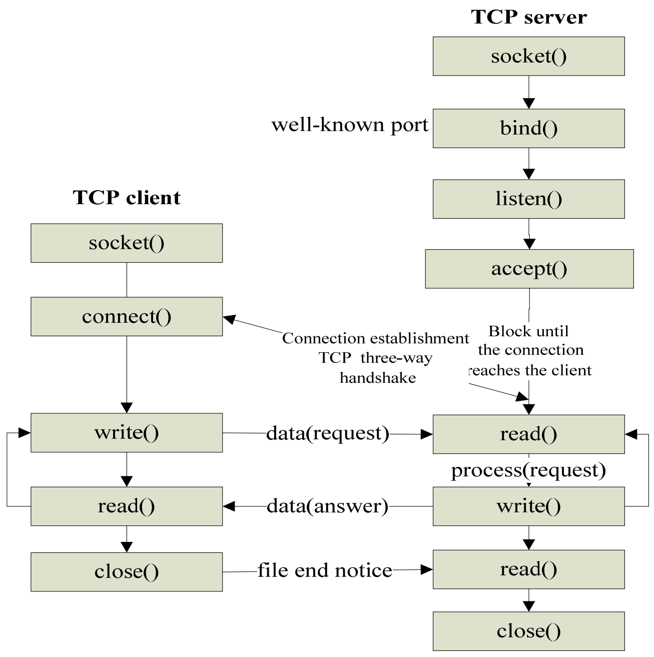

Fig. 2. Basic TCP/IP Socket Functions

RS-232 is a standard serial interface. It is commonly used in computers, peripherals, switches and many other communication equipments. Linux provides good support on RS-232 from the beginning.

Serial files of Linux are /dev/ttyS0 and /dev/ttyS1 in /dev. The basic operations include opening serial port, setting serial port, reading serial port, writing serial port and closing serial port. The basic serial port settings includes baud rate setting, check bit setting and stop bit setting. The serial port settings will mainly set the various member values of structure struct termios:

```
Struct termios
{
unsigned short c_iflag;     /* input pattern flag */
unsigned short c_oflag;     /* output pattern flag */
unsigned short c_cflag;     /* control pattern flag */
unsigned short c_lflag;     /* local pattern flag */
unsigned char c_ine ;       /*linediscipline*/
unsigned char c_cc[NCC]:    /* special control character */
}
```

The five members in termios structure correspond to the input, output, control, local patterns and special characters of terminals.

The embedded serial communication of Linux is illustrated as follows:

Step1. Open the device file by OPEN. Open flag O_NONBLOCK and O_NOCCTY will be used. O_ONBLOCK will return immediately without waiting, and O_OCCTY specifies the serial device is not control terminal;

Step2. Set communication baud rate by cfsetispeed and cfsetospeed;

Step3. Set miraculous bit;

Step4. Read or write serial port;

Step5. Read or write end, and close the serial port by CLOSE.

III. HOME GATEWAY DATABASE DESIGN BASED ON SQLITE

A. *SQLite Architecture*

The mainly 8 subsystems of SQLite are shown in Fig 3, which is similar to the relational database management. There are mark processor (tokenizer) and analyzer (parser) at the top of the figure. SQLite has its highly optimized code generator, which can create highly efficient code rapidly.

The bottom is based on Knuth optimized B-tree, which makes it run on adjustable page cache and the search of the disk minimized. Further down the figure is the page cache, which acts on the OS abstraction layer to help move the database. The core architecture is a virtual database engine (VDBE). VDBE completes all the operations related to the data, and acts as the center unit of information exchange between clients and storages. From every point of view, it is the heart of SQLite. After analyzing the SQL statement, VDBE begins to work. Code generator will translate the analysis tree into a pocket program, and then these procedures are combined into a series of instructions expressed by a virtual machine with VDBE language. So forth, VDBE executes each instruction, and fulfills the final query specified by SQL statement [9].

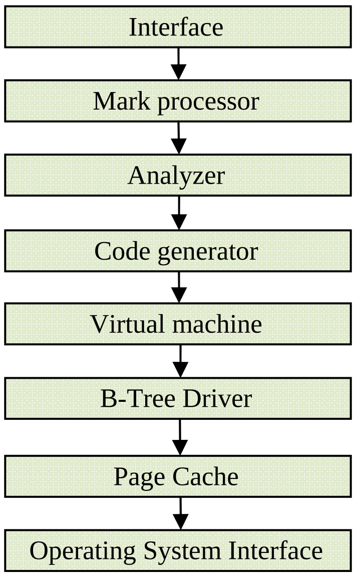

Fig. 3. SQLite Architecture

SQLite realization on Linux

The API of SQLite is quite easy to be used, which has only three functions for executing SQL and obtaining data. It is extendable which allows programmers to customize functions and integrate by callback. Programming C language API is the basis of other interfaces. The open source community has extended a large number of customer interfaces, which makes it impossible for SQLite using by other languages.

First database sysdb and table user are created by sqlite command lines or image tools such as sqlite browse, and so on. It makes that the database can be connected by calling sqlite_open() using the parameters such as file names and visiting modes. For example, “p_db=sqlite open("./sysdb", 0777, 0);”. For executing a SQL query, sqlite_exec() can be called, such as “sqlite_ exec(p_db,"select*fromuser; ",callback,0,0);”. For SQLite executes callback function for each record to obtain results from database, so own callback function can be created for realizing detailed function. Finally, statement "sqlite_ close (p_ db); " is used to close database.

B. Database Design

SQLite database adopts file mode, so we create a db named database file and plan necessary data tables based on project requirements.

The database SQL generated is as following:

BEGIN TRANSACTION;

create table device(sys_id integer(2) primary key, name char(20), alias char(20),

sort integer(2), subsort integer(2), reg_ date datetime);

create table devicesort(sort integer(2) primary key, sort name char(20));

create table devicesubsort(subsort integer(2) primary key, sort integer(2), sortname char(20)) ;

create table sendcom(sys_ id integer(2), com_code char(8), com_par integer(8),

com_string char(24)) ;

create table user(username char(10), password char(8), type char(8));

create table hiscom(sys_id integer(2), com_string char(24), com_out char(32));

create table com0(com_code char(8), com_name char(10), com_par integer(8), unit char(10), pares low integer(8), par lar integer(8));

create table stat0(com_code char(8), com_name cha(10), com_par integer(8), unit char(10)) ;

create table coml(com_code char(8),com_name cha(10), com_par integer(8),

pares_low integer(8), par_high integer(8), unit char(10_par integer(8),unitchar(10));

COMMIT;

The data tables include mainly device (system device table), sort (system device sort table), subsort (system device subsort table). The device table is used for saving the registered system devices; sort and subsort tables are used to distinguish categories of equipment (home appliances, security, three meters) and subcategories in a single category

(such as home appliances can be subdivided into washing machines, refrigerators, etc.)

The data tables related to device control are com (equipment order table), stat (device status table). The com table is used to store all the control commands for some device, while the stat table is used to keep real-time status of the device. Tables related to communications are mainly sendcom (outgoing command table) and hiscom (sent command table). The sencom table is used to store control commands collected from Web pages, local client and telephone terminal control commands. In order to reduce the number of database operations, various commands which have been sent in the past can be obtained directly from hiscom table to get higher implementation efficiency. Other auxiliary data tables such as user (user table) are used for the control of read and write permissions to enhance the database security.

IV. TCP/IP PROTOCOL IN HOME GATEWAY

A. TCP/IP Protocol Architecture

TCP/IP (Transmission Control Protocol/Internet Protocol) was originally developed by the U.S. DARPA (Department of Defense Advanced Research Project Agency, DARPA) in 1969 and applied in the ARPANET (Advanced Research Project Agency). With the growth on ARPANET scale and scope, TCP/IP protocol has gradually been the basis of Internet, and its application range is also more and more widely. It has almost become the standard network protocols of WAN and LAN: TCP/IP protocol suite put forward the hardware vendor-independent data transfer format and rules.

Because of its unique hardware independence, it has been accepted by many systems so quickly.

UNIX uses TCP/IP protocol, Windows NT and Novell's NET ware has also software or modules to support TCP/IP. The reason that is called as TCP/IP protocol suite is that TCP/IP includes a series of protocols and applications [10].

TCP/IP protocol suite complies with a four-layer reference model including the interface layer, network layer, transport layer and application layer. Its architecture is shown in Fig 4.

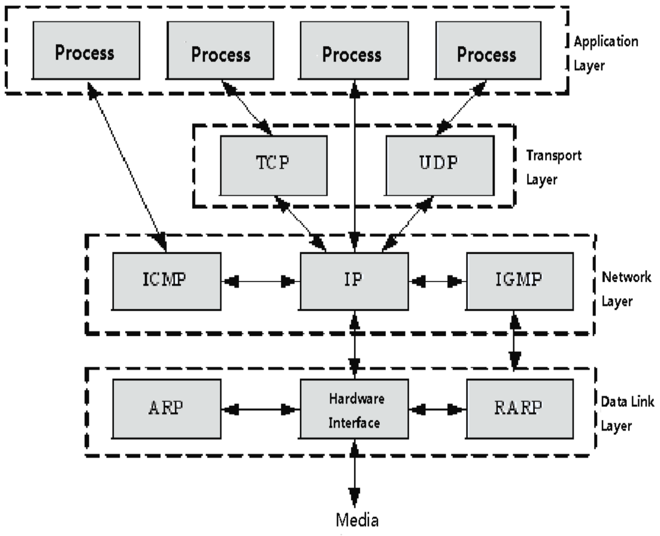

Fig. 4. TCP/IP Architecture

TABLE I
COMPARISON WITH TCP / IP REFERENCE MODEL AND OSI REFERENCE MODEL

Layer in the OSI	Function	TCP / IP protocol suite
Application Layer	File Transfer, E-mail, File Service, Virtual Terminal	TFTP，HTTP，SNMP，FTP，SMTP，DNS，RIP，Telnet
Presentation Layer	Data Format, Code Conversion, Data Encryption	No protocol
Session Layer	Remove/Create links with other nodes	No protocol
Transport Layer	Provides end-to-end interface	TCP，UDP
Network Layer	Select route for packets	IP，ICMP，OSPF，BGP，IGMP，ARP，RARP
Data Link Layer	Transmit Frames with Address and Error Detection	SLIP，CSLIP，PPP，MTU
Physical Layer	Transmit Data in Binary Format on Physical Media	ISO2110，IEEE802，IEEE802.2

B. Comparison with TCP / IP Four-layer Reference Model and OSI Seven-layer Reference Model

The interconnection reference model of traditional open systems is a seven-layer abstract reference model of communication protocols, in which each layer performs a specific task. The TCP / IP protocol uses four-layer hierarchy,

and each layer calls the network provided by the next level to complete its requirements. The comparison with OSI Reference Model and TCP / IP reference model are shown in TABLE I.

C. TCP/IP Protocol Stack Model in Home Gateway

TCP/IP protocols stack needs to be written with software by their own. Taking into account the specific circumstances of system protocol, the protocols are simplified greatly. The realization of the simplified gateway protocol layer structure is shown in Fig 5.

HTTP in application is to implement layer gateway as a WEB server. In order to achieve easy access to remote terminal, HTTP protocol is used in the application layer.

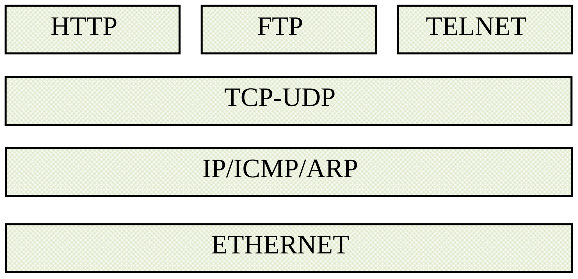

Fig. 5. TCP/IP Protocol Stack in Home Gateway

This way, the gateway can respond to the request sent from client browsers. The gateway responses request and return webpage information to client browser. CGI programs coded on server can realize the data exchange between clients and server.

FTP is a file transfer protocol which is mainly used as downloaded file storage of the gateway. It can achieve the information update.

TELNET is the most widely used application in Internet. We can log (register) on a remote host, and then log on any other host by network without connecting hardware terminal for each host (of course a login account is needed). Telnet client process also interacts with end user TCP/IP protocol modules. Usually, any information transmission is connected by TCP, and any returned information in the connection will be output to the terminal.

Transport layer adopts hybrid TCP-UDP data for information transfer.

This way can reduce TCP session time, increase the data rate, and provide reliable service for the HTTP application layer. Because TCP is connection-oriented implementation, reliable data transmission can be guaranteed but connection and session have to be established for each communication and then disconnect. Therefore, when the transmission is just small amount of data, this way is too much overhead. UDP communication is non-oriented, so a combination of both is applied on the realization of the transport layer. When reliable transmission is needed, TCP approach is used; when only a small amount of data needs to be transferred, UDP mode is applied. By this way, the whole data communication efficiency can be improved.

IP realizes the detection for data packet checksum. Once the checksum has errors, the data packet will be discarded, and ICMP messages will be reported to the source side.

ICMP response to PING request for debugging, and test the network coherency.

ARP is the Address Resolution Protocol. When the received Ethernet packet header is 0X0806, ARP receives data packets and detects whether the checksum is 0. Once the condition is found, the processor will check whether it is a IP address request or response. If it is a response, MAC address will be stored in the Ethernet IP address table. The table reserves the last 10 MAC addresses of gateway server communication. And, if a request is received, the processor will return their address to the other side.

V. WEB SERVER REALIZATION

From the functions in principles, Web servers need to listen for client service requests and provide appropriate services according to the type of user requests. Clients use Web browsers and Web servers to take communications. After the Web server receives a request form the client, it will handle user requests and return the required data. CGI (Common Gateway Interface) provides an external program running access for Web server, which enables the interaction between browser and server. CGI program is an external program, which needs to be compiled into an executable file for running in the server. The application structure is shown in Fig 6.

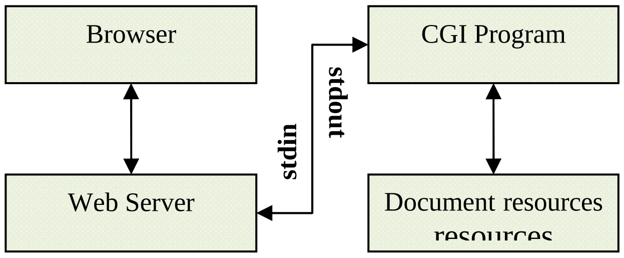

Fig. 6. CGI Function Diagram

Browser sends the user input data to Web server and the Web server send STDIN used by data to CGI programs. After CGI is executed, database records might be accessed and output result file with HTML forms by using STDOUT. The result can be displayed for clients on browser sent by Web server.

Master process of the server monitors the set port. Once the browser request arrives, a connection will be establishes and new socket descriptor is returned to the sub-process. The sub-process reads the request, decomposes URL and request methods, and then determines according to MIME types corresponding to the requested file name extension. If these are static text, read directly and send them to the browser; if these are CGI scripts, a new sub-process will be created to execute the scripts, the running results will be processed and returned to the browser. If no follow-up request in a certain time, the connection will be closed.

Open sourced embedded Web server Boa is adopted.

A. Boa Source Codes Are Downloaded from www.boa.org, and Decompressed into src Subdirectory of Source Code Directory

```
#tar -xzf boa-0.94.13.tar.gz
```

```
#cd boa-0.94.13/src
```

Generate Makefile file.

```
#./configure
```

Modify Makefile file, find CC=gcc and CPP=gcc –E, change them to the directories CC=/opt/host/bin/arm-linux-gcc and CPP=opt/host/bin/arm-linux-gcc-E which the cross compiler is installed. Save and exist, then run make to compile to get the executable program boa.

```
#make
```

```
#/opt/host/bin/arm-linux-strip boa
```

B. Boa Configuration

Boa needs to create a boa directory in /etc directory on virtual machine and put the main configuration file boa.conf into Boa. In Boa source directory, there has already an example boa.conf based on which can be modified.

Group nogroup is modified as group 0. Because there has no nogroup in file /etc/group, so it is set as 0. And there has nobody user in /etc/passwd, so user nobody need not be modified.ScriptAlias/cgi-bin//usr/lib/cgi-bin/ is modified as ScriptAlias/cgi-bin//var/www/cgi-bin/, and others are default configuration.

Log file needs to be created in /var/log/boa. The main directory for creating HTML documents is /var/www, and stores static webpages into this directory. CGI scripts are generated in /var/www/cgi-bin,mand cgi scripts are stored in this directory. And file mime.types is copied to //etc directory. Generally, it can be copied directly from /etc directory of Linux host.

VI. CONCLUSION

Embedded Linux technology, embedded database, home gateway protocol stack and embedded Web server principles are studied; embedded database SQLite and embedded Web server Boa are successfully applied on home gateway; home gateway database is designed to achieve the operating system migration, basic software function support for embedded home gateway and intelligent home appliances.

The most extensive areas of computer applications are embedded technologies. In all IT products, the embedded market will be huge. Although the development of embedded systems has been several decades of history, this is only the beginning. We believe, Linux which has many technical advantages is bound to be increasingly widely used in embedded systems. Embedded Linux will be an important part in embedded systems.

ACKNOWLEDGMENT

Here, we give thanks to our professions who gave us support and advice, especially Professor Huamin Yang in School of Computer Science and Technology of Changchun University of Science and Technology.

REFERENCES

[1] ZHANG Litao. “Design of Digital Home Network System Architecture Based on Intelligent Gateway”, *Modern Electronic Technique*, vol.320. pp: 84-86, 2010.

[2] CHENG Zhongfang. “Development and System Structure of Home Gateway”, *Computer Knowledge and Technology*, vol.4. pp: 551-552, 2008.

[3] LIU Li, LU Yang, OUYANG Xin, SHUAI Chun yan. “Research and implementation of embedded open framework based on electronic household appliances”, *Computer Engineering and Design*, vol.27, pp: 232-234, 2006.

[4] Liu Huawei, Liu Quan, Luo Yaohua. “Design of intelligent information household appliances embedded in the internet”, *Applied Science and Technology*, vol. 128.pp: 31-33, 2001.

[5] Michael Owens. “Embedding an SQL Database with SQLite”, *Linux Journal*,June, vol. 110. pp: 62-64, 2003.

[6] Tan T.K, Raghunathan A, Jha N.K. EMSIM. “An energy simulation framework for an embedded operating system”, *IEEE International Sysmposium on Circuits and Systems,* vol.2. pp: 46-47, 2002.

[7] SUN Yuhong, CUI Shaobin. “Research on embedded server of home network”, *Computer Engineering and Design*, vol.29. pp: 1394-1395, 2008.

[8] Guomei Liu, Anping Zheng. “The implementation of dynamic Web technology in uClinux”, *Microcontrollers & Embedded System*s, vol.2. pp: 67-69, 2004.

[9] Klaus Wehrle, Frank Pahlke, Hartmut Ritter, Daniel Muller, Marc echle. “The Linux® Networking Architecture: Design and Implementation of Network Protocols in the Linux Kernel”,Prentice Hall Publisher. 2004, pp: 99-104.

[10] CHEN Xin, GUO Lihao, TANG Zhenzhou. “Design of network appliance gateway based on TCP/IP protocol stack”, *Computer Engineering and Design*, vol.28. pp: 1236-1238, 2006.